Ultra-low-power RRAM-based FPGA: A Road towards Reconfigurable Edge Computing

Xifan Tang*, Edouard Giacomin*, Natan Chetrit †, Pierre-Emmanuel Gaillardon*†
* Department of Electrical and Computer Engineering, University of Utah †ReRouting, LLC
Email: {xifan.tang|edouard.giacomin|pierre-emmanuel.gaillardon}@utah.edu, natan@reroutingllc.com

***Abstract*—The trends in *Internet-of-Thing* (IoT) require specialized hardware systems to be more computing capable than ever while at the same time satisfying an ultra-low power budget. *Field Programmable Gate Arrays* (FPGAs), thanks to their reconfigurable nature, have been an ubiquitous media in many edge computing systems. However, low-power FPGAs generally suffers from large delay degradation (up to 2×), missing to achieve the computing capability required by many modern edge computing applications. In this paper, we investigate the opportunity of using *Resistive Random Access Memories* (RRAMs) in ultra-low-power FPGA architectures. We (i) evaluate the circuit design aspects of RRAM-based routing multiplexers; (ii) introduce a novel design flow to accurate analyze FPGA architectures; and (iii) study the opportunity of building near-V_t RRAM-based FPGA. Full-chip layouts and SPICE simulations present that at nominal operating voltage, RRAM-based FPGAs can improve up to 8%/22%/16% in area/delay/power, as compared to SRAM-based counterparts. Compared to SRAM-based FPGAs working at its nominal voltage, a near-V_t RRAM-based FPGAs can outperform by about 2 × the *Energy-Delay Product* without delay overhead.**

***Keywords*—FPGA; Resistive Memory; Low-power Device;**

I. Introduction

The trends in *Internet-of-Thing* (IoT), such as edge computing, require specialized hardware systems to be more computing capable than before while at the same time satisfying an ultra-low power budget. *Field Programmable Gate Arrays* (FPGAs), thanks to their reconfigurable nature, have been a ubiquitous media in many edge computing systems. However, the expensive configurable routing architecture, which accounts for about 70% of the area, 80% of the delay and 60% of the power of the whole chip [1], is preventing them to achieve ultra-low energy efficiency. This leads FPGAs to experience approximately 20× bigger area, 4× longer delay, and 12× higher power consumption compared to *Application-Specific Integrated Circuits* (ASICs) [2]. In particular, the power consumption is a serious barrier for the distribution of FPGAs in a large set of IoT applications. Previous works [3], [4] demonstrate low-power FPGA designs where a low supply voltage is employed to save up to 50% of the power consumption. However, low-power FPGAs generally suffers from large delay degradation (up to 2×), missing to achieve the computing capability required by edge computing.

Resistive Random Access Memory (RRAM) technology, a member of *Non-Volatile Memory* (NVM) family, opens the opportunity in advancing FPGA technologies towards IoT applications by bringing non-volatility and performance enhancements [5]–[9]. A RRAM device can be treated as a reconfigurable resistor, which can be switched to either *High Resistance State* (HRS) or *Low Resistance State* (LRS) thanks to application of different combinations of programming voltage and current. Being *Back-End-of-Line* (BEoL) compatible, RRAM technology allows the configuration memories of FPGAs to be fabricated on the top of transistors, thereby increasing the integration density and shortening the metal interconnections. As depicted in Fig. 1, non-volatility of RRAMs could bring significant power reduction benefits considering their zero leakage power in sleep mode. Even more tantalizing, major works focus on proposing novel programmable switches with the objective of replacing a *Static Random Access Memory* (SRAM) and a transmission-gate with a unique RRAM device [7]–[9]. With lower resistance R_{LRS} than transistors and also smaller parasitic capacitances, RRAMs can bring remarkable improvements on the delay and power to routing multiplexers. Previous works predicted that these proposed RRAM FPGAs can improve area by 7-15%, delay by 45-58% and power by 20-58%, when compared to SRAM-based counterparts [7]–[9]. Furthermore, as R_{LRS} of RRAMs are independent from operating V_{DD}, this opens the door to energy efficient edge computing at near-V_t regime without any performance loss [10].

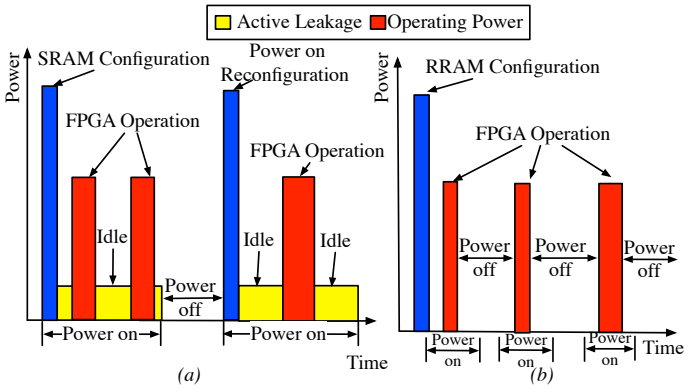

Fig. 1. Power consumption of (a) a SRAM-based FPGA and (b) a RRAM-based FPGA.

Pierre-Emmanuel Gaillardon, Natan Chetrit and Xifan Tang have financial interests in the company ReRouting LLC, which manufactures RRAM-based systems and provides engineering service.

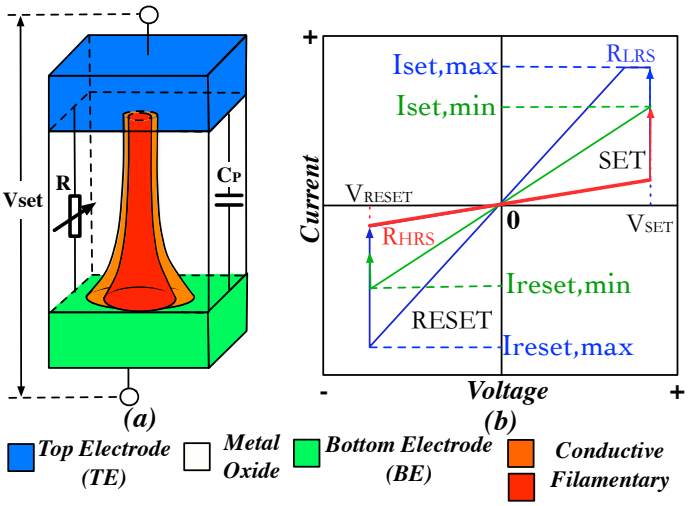

Fig. 2. RRAM structure: (a) size of filaments inside a RRAM achieved by $I_{set,min}$; (b) Size of filaments inside a RRAM achieved by $I_{set,max}$; (c) I-V characteristics of a RRAM with Bipolar Resistive Switching.

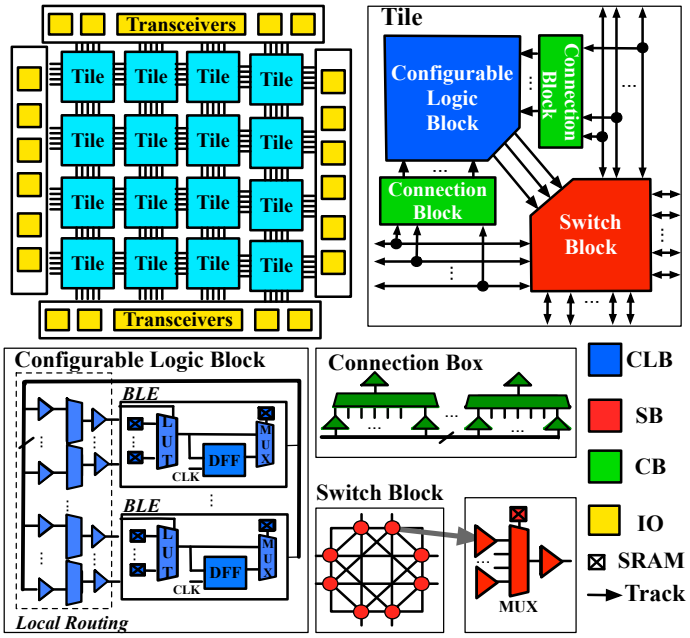

Fig. 3. General FPGA architecture.

In this paper, we (i) evaluate the circuit design aspects of RRAM-based routing multiplexers; (ii) introduce a novel design flow to accurate analyze FPGA architectures; and (iii) study the opportunity of fabricating near-V_t RRAM-based FPGAs. Full-chip layouts and SPICE simulations present that at nominal operating voltage, RRAM-based FPGAs can improve up to 8%/22%/16% in area/delay/power, as compared to SRAM-based counterparts. Compared to SRAM-based FPGAs working at its nominal voltage, a near-V_t RRAM-based FPGAs can outperform by about 2 $\times$ the *Energy-Delay Product* without delay overhead.

The rest of this paper is organized as follows: Section II introduces necessary background knowledge about RRAM technology and FPGA architectures. Section III explains the evaluation methods developed for RRAM-based circuits and FPGAs. Section IV presents the circuit results about RRAM-based multiplexers. Section V shows the architecture-level study on the near-V_t RRAM-based FPGAs. Section VI concludes the paper.

II. Background

In this section, we provide necessary background knowledge related to the topic. We first introduce the RRAM technology and then give a brief overview on current state-of-the-art FPGA architectures.

A. RRAM technology

Resistive Random Access Memories (RRAMs), a promising emerging memory technology [11], typically consists of three layers: a *Top Electrode* (TE), a transition metal oxide material stack and a *Bottom Electrode* (BE), as highlighted in Fig. 2(a) [12]. Thanks to its compatible with *Back-End-of-the-Line* (BEoL), a RRAM can be freely fabricated anywhere between two metal layers on the top of transistors, leading to a high integration density.

Through a filamentary conduction mechanism in the metal oxide layer, RRAMs can be switched between two stable resistance states: the *High Resistance State* (HRS) and the *Low Resistance State* (LRS), by applying a programming voltage across the TE and BE. The switching event from LRS to HRS is called **set** process, while the opposite one is called **reset** process. In this paper, we consider RRAM based on *Bipolar Resistive Switching* (BRS) only, which is a common choice for most RRAM-based circuits and systems [7]–[10], [13]–[17]. Fig. 2(c) illustrates the I-V characteristics of a BRS RRAM. The minimum programming voltages required to trigger **set** and **reset** processes are defined as V_{set} and V_{reset}, respectively. The programming currents that are supplied during the set and reset processes are defined as I_{set} and I_{reset}, respectively. A current compliance on I_{set} is often enforced to avoid a permanent breakdown of the device, which is denoted by $I_{set,max}$ in Fig. 2(c). The programming current tunes the size of filaments, leading to a difference in the resistance of a RRAM in LRS, R_{LRS}. Take the examples in Fig. 2-(a) and (b), the filament highlighted in orange leads to a lower R_{LRS} than the filament highlighted in red.

More details about RRAM technology can be found in [12].

B. Conventional FPGA technology

Conventional FPGAs consist of an array of tiles surrounded by IO blocks, as illustrated in Fig. 3. Each tile contains a *Configurable Logic Block* (CLB), a *Connection Block* (CB) and a *Switch Block* (SB) [18]. A CLB consists of N *Basic Logic Elements* (BLEs) and a local routing architecture providing inner-block interconnections. A BLE contains a *Look-Up Table* (LUT), a *Flip-Flop* (FF) and a 2:1 multiplexer, which selects either a combinational or a sequential output. SBs interconnect routing tracks between tiles, while CBs connect

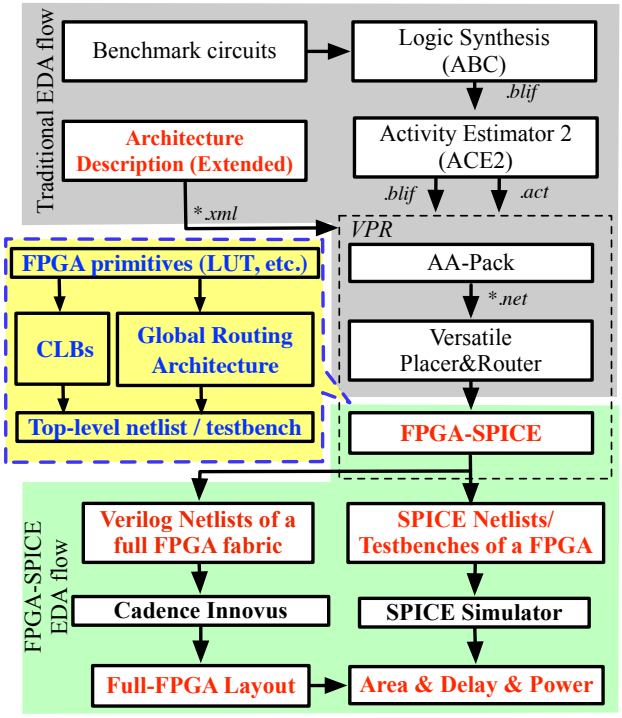

Fig. 4. EDA flow used for RRAM-based and SRAM-based FPGA architecture evaluation.

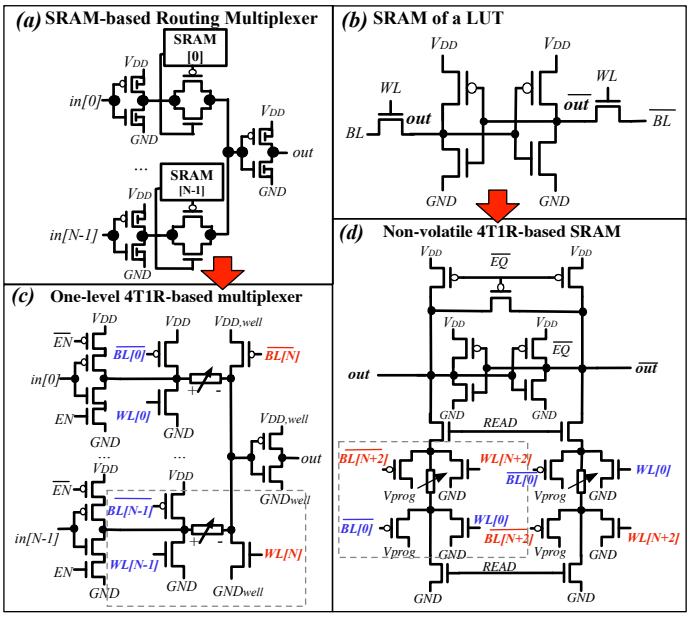

Fig. 5. Circuit designs of (a) SRAM-based routing multiplexer; (b) 4T1R-based routing multiplexer; (c) SRAM; (d) Non-volatile 4T1R-based SRAM.

routing tracks to CLB input and output pins inside a tile. To accelerate arithmetic-intensive applications, commercial FPGAs [19]–[21] adopt various architectural enhancements, such as fracturable LUTs [22], hard carry chains and heterogenous blocks. As we aim at capturing the difference between SRAM-based and RRAM-based FPGAs, we consider, without the loss of generality, the homogenous tile-based FPGA architecture shown in Fig. 3 in this paper.

III. FPGA-SPICE: An Open Source Flow for Accurate FPGA Analysis

We developed an open-source flow to perform accurate area, delay and power analysis on FPGA architectures. As illustrated in Fig. 4, the new flow consists of two parts:

1) The traditional VPR-based FPGA EDA flow [23], where benchmark circuits are logic optimized by ABC [24] and then processed through activity estimation [25], packing, placement and routing of original VPR [23].
2) A novel EDA add-on capable of modeling a full FPGA fabric in Verilog and SPICE netlists from the VPR architecture description:
   - To enable accurate area analysis, we have developed a synthesizable Verilog generator for both SRAM-based and RRAM-based FPGAs, with which layouts of full FPGA fabrics can be derived by employing a semi-custom design flow. Note that in addition to area results, the full-chip layout can be directly used for fabrication purpose, enabling fast prototyping for both SRAM-based and RRAM-based FPGAs.
   - To perform delay analysis, we run SPICE simulations for each component in a FPGA, i.e., LUTs, FFs and multiplexers. The timing results are back-annotated to the timing analysis engine in VPR to estimate accurate critical path delays.
   - To enable an accurate power analysis, we enhanced FPGA-SPICE [26], [27] to output SPICE netlists modeling RRAM-based circuits and FPGA architectures [10]. HSPICE [28] is employed to perform power analysis and total power consumption is achieved by summing up the power results extracted from each HSPICE simulation.

IV. Using RRAMs to Build Innovative Routing Multiplexer Design

In this part, we evaluate the circuit design aspects of RRAM-based multiplexers based on the current state-of-art 4T(ransistor)1R(RAM) programming structure [10] as shown in Fig. 5(c) by comparing to their CMOS counterparts in Fig. 5(a).

A. Methodology

Both CMOS and RRAM-based multiplexers are designed using a commercial 40nm technology. The datapath circuits and the 4T1R programming structures are built with standard logic transistors ($W/L = 140nm/40nm$). CMOS multiplexers employ transmission gates, which are implemented by a pair of minimum-width *n*-type and *p*-type logic transistor. Input and output inverters are sized to 3× minimum width in order to resist the parasitics of metal wires. To match our FPGA architecture assumption in Section III, the input sizes of multiplexers are swept from 2 to 50.

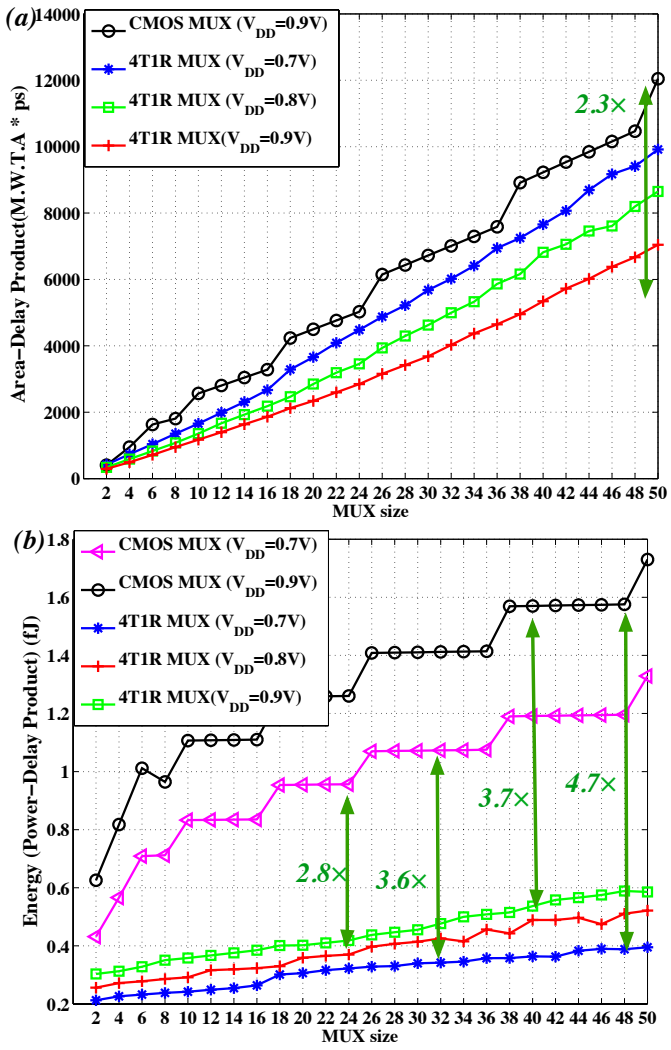

Fig. 6. Area-Delay product and Power-delay product comparison between SRAM-based and RRAM-based multiplexers operating at nominal and near-V_t regime.

In the rest of this paper, we consider a RRAM technology [29] with programming voltages $V_{set} = |V_{reset}| = 1.1V$ and a maximum current compliance of $I_{set} = |I_{reset}| = 500\mu A$. The lowest achievable R_{LRS} of a RRAM is $2.2k\Omega$ while the R_{HRS} is $20M\Omega$ in order to guarantee a good energy efficiency [16], [30]. The Stanford RRAM compact model [31] is used to model the considered RRAM technology. To accurately include the parasitic effects from the co-integration, we add a parasitic capacitance of $13.2aF$ to the RRAM SPICE model, which is estimated by considering the height and the dimension of metal vias in the commercial 40nm technology.

B. Area, Delay and Power Efficiency

To explore the inherent trade-offs with area, delay and power, we compare in Fig. 6 the *Area-Delay Product* (ADP) and *Power-Delay Product* (PDP) of CMOS and RRAM-based multiplexers operating at nominal and near-V_t regime. Thanks to BEoL integration and lower R_{LRS} than transistors, *Area-Delay Product* (ADP) of 4T1R-based multiplexers can be up to $2.3\times$ more efficient than CMOS multiplexers than CMOS multiplexers, as illustrated in Fig. 6(a). Due to the fact that resistance of RRAM is independent from V_{DD}, *Power-Delay Product* (PDP) of 4T1R-based multiplexer improves over $4.7\times$ the one of CMOS multiplexers, as shown in Fig. 6(b). $V_{DD} = 0.7V$ guarantees the best PDP, in other words energy efficiency, for 4T1R-based multiplexers.

In summary, 4T1R-based multiplexers are more efficient in area, delay and power at both nominal V_{DD} and near-V_t regime than CMOS multiplexers. In particular, such energy reduction is achieved along with significant delay improvements.

V. Low-Power RRAM-based FPGA

In this part, we first introduce the generality of our RRAM-based FPGA architecture and experimental methology to evaluate FPGA performance. Then, in Section V-C and V-D, we study the area and power characteristics of the proposed RRAM-based FPGAs using full-chip layouts and electrical simulations.

A. Vision on RRAM-based FPGA Architecture

We propose a near-V_t RRAM-based FPGA where the conventional SRAM-based primitive blocks (Fig. 5(a)(b)) are replaced by RRAM-based circuits (Fig. 5(c)(d)). To achieve non-volatility, all the SRAM-based circuits in FPGA architectures are replaced with RRAM-based implementations. We apply two different strategies depending if we are replacing the SRAMs of routing multiplexers or LUTs.

1) The whole SRAM-based routing multiplexers are replaced by 4T1R-based counterparts, as illustrated in Fig. 5(a) and (c). We borrow the 4T1R-based routing multiplexer designs from [17], where both SRAMs and transmission-gates are replaced by 4T1R elements. Hence, RRAMs behave not only as memory cells but also as logic gates that propagate or block datapath signals. Thanks to the low R_{LRS} and efficiently sharing programming transistors, the 4T1R-based routing multiplexers can bring significant improvements in area, delay, power and especially in energy consumption [17]. More importantly, such replacement leads to the performance improvements without challenging the the endurance limit of RRAM devices. An actual programming operation for 4T1R-based multiplexers occurs infrequently, only during FPGA reconfiguration.
2) In LUTs, only the SRAMs are replaced by RRAM-based non-volatile SRAM topology, as illustrated in Fig. 5(b) and (d). Different from routing multiplexers, the *on/off* state of datapath transistors can be switched frequently during each operating cycle. Note that the data storage of SRAMs is changed only during reconfiguration, which has a low switching rate tolerable to RRAM endurance. Therefore, for LUTs, RRAMs are used to grant non-volatility to SRAMs, rather than to datapath transistors.

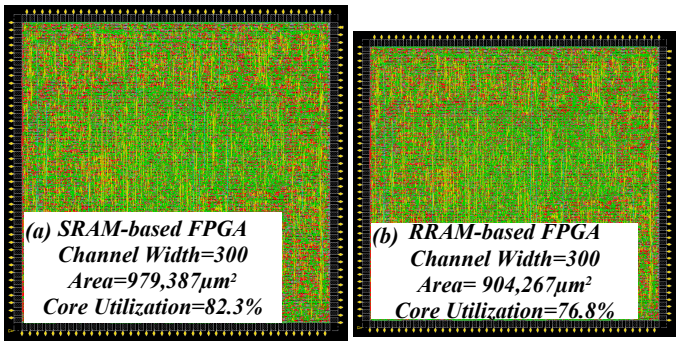

Fig. 7. Full-chip layouts (Channel width is set to 300) of FPGAs configured by BL and WL decoders: (a) SRAM-based and (b) RRAM-based.

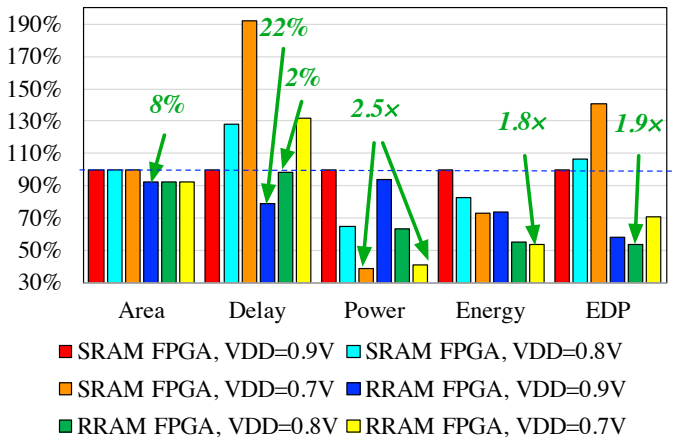

Fig. 8. Area, delay and energy comparison between SRAM-based and RRAM-based FPGAs operating at nominal and near-V_t regime.

B. Evaluation Methodology

To provide a fair comparison, both SRAM-based and RRAM-based FPGAs employ a CLB architecture with forty inputs pins ($I = 40$). Each CLB consists of ten BLEs ($N = 10$), each of which contain a 6-input LUT ($K = 6$) [32]. Similar to commercial FPGAs [19], [20], we consider unidirectional routing architectures [33] with three types of wire lengths. In each routing channel, 30% of routing tracks are built with length-1 wires ($L = 1$), another 30% of routing tracks are built with length-2 wires ($L = 2$) and the rest 40% of routing tracks are built with length-4 wires ($L = 4$). Each routing track can be connected to other three routing tracks from adjacent channels ($Fs = 3$). Each CLB input pin can be connected to 15% of the routing tracks in a channel ($F_{c,in} = 0.15$), while each CLB output pin can reach 10% of the routing tracks ($F_{c,out} = 0.10$).

Both SRAM-based and RRAM-based FPGAs are built with the same commercial 40nm technology and RRAMs used in Section IV for the multiplexer evaluation. To guarantee the best overall performance, CMOS multiplexers in local routing architecture and CBs adopt a two-level structure while the others are built with a one-level structure [32], [33]. All the RRAM-based multiplexers adopt a one-level structure and RRAMs are placed between the first and the second metal layer, for best overall performance [17].

We exploit the novel EDA flow in Fig. 4 to compare the area, delay and power of SRAM-based and RRAM-based FPGAs. The twenty largest MCNC benchmarks [34] are selected as the input of the EDA flow. All the experiments are run on a 64-bit RedHat Linux server with 28 Intel Xeon processors and 256GB memory.

C. Area Characteristics

Fig. 7 presents the full-chip layouts of SRAM-based and RRAM-based FPGAs, both of which including core logics, configuring peripherals and IOs. Note that both FPGAs contain a channel width of 300, which is similar to commercial FPGAs [19], [20]. For sake of the capability of our workstation, we consider a CLB array size of 5×5 which are surrounded by 160 I/O pads. Note that the achieved area results with a 5×5 CLB array can be representative because large FPGAs can be regarded as an assembly of the small CLB arrays. The full-chip layout comparison shows that RRAM-based FPGAs brings a 8% area reduction when compared to the SRAM-based FPGA counterpart, thanks to the compactness of RRAM-based multiplexers.

D. Near-V_t Energy efficiency

Fig. 8 compares the proposed RRAM-based FPGAs to a well-optimized SRAM FPGA. Considering the nominal $V_{DD} = 0.9V$, RRAM-based FPGAs can improve up to 8% in area, on average 22% in delay and on average 16% in power respectively, as compared to SRAM-based counterparts. Even when V_{DD} is reduced to near-V_t regime, i.e., $0.8V$, RRAM-based FPGA remains the same performance-level as the SRAM-based FPGA at nominal voltage. This is due to the resistance of RRAMs being independent from working voltage, unlike transistors whose equivalent resistance degrades seriously at near-V_t regime. When operating at near-V_t regime, RRAM-based FPGAs can improve *Energy-Delay Product* by close to $2\times$, as compared to SRAM-based FPGA operating at nominal working voltage. Note that the energy reduction is achieved without any delay overhead.

VI. Conclusion

In this paper, we first evaluated a novel routing multiplexer design based on RRAMs. SPICE simulations showed the multiplexers can reduce the energy consumption by $4.7\times$ without any performance loss, when compared to well-optimized CMOS counterparts. We then introduce an open-source EDA flow based on FPGA-SPICE, which can autogenerate SPICE and Verilog netlists of full FPGA fabrics, enabling accurate evaluation and rapid-prototyping. Using the circuit-level results and FPGA-SPICE, we present a near-V_t RRAM FPGA architecture that can outperform standard SRAM-based FPGAs by close to $2\times$ in *Energy-Delay Product* without delay overhead, unlocking a more energy-efficient and reconfigurable edge computing paradigm.

ACKNOWLEDGMENT

This material is based on research sponsored by Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under agreement number FA8650-18-2-7855. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) or the U.S. Government.

REFERENCES

[1] M. Lin, A. E. Gamal, Y. C. Lu, and S. Wong, "Performance benefits of monolithically stacked 3-d fpga," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 216–229, Feb 2007.

[2] I. Kuon and J. Rose, *Quantifying and Exploring the Gap Between FPGAs and ASICs*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[3] T. Tuan, A. Rahman, S. Das, S. Trimberger, and S. Kao, "A 90-nm low-power fpga for battery-powered applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 296–300, Feb 2007.

[4] B. H. Calhoun, J. F. Ryan, S. Khanna, M. Putic, and J. Lach, "Flexible Circuits and Architectures for Ultralow Power," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 267–282, Feb 2010.

[5] Y. C. Chen, W. Wang, H. Li, and W. Zhang, "Non-volatile 3D stacking RRAM-based FPGA," in *22nd International Conference on Field Programmable Logic and Applications (FPL)*, Aug 2012, pp. 367–372.

[6] K. Huang, R. Zhao, W. He, and Y. Lian, "High-Density and High-Reliability Nonvolatile Field-Programmable Gate Array With Stacked 1D2R RRAM Array," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 1, pp. 139–150, Jan 2016.

[7] S. Tanachutiwat, M. Liu, and W. Wang, "FPGA Based on Integration of CMOS and RRAM," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 11, pp. 2023–2032, Nov 2011.

[8] J. Cong and B. Xiao, "FPGA-RPI: A Novel FPGA Architecture With RRAM-Based Programmable Interconnects," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 4, pp. 864–877, April 2014.

[9] P.-E. Gaillardon, D. Sacchetto, S. Bobba, Y. Leblebici, and G. D. Micheli, "GMS: Generic Memristive Structure for Non-Volatile FPGAs," in *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC)*, October 2012, pp. 94–98.

[10] X. Tang, G. Kim, P.-E. Gaillardon, and G. De Micheli, "A Study on the Programming Structures for RRAM-based FPGA Architectures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 4, pp. 503–516, 2016.

[11] G. W. Burr, B. N. Kurdi, J. C. Scott, C. H. Lam, K. Gopalakrishnan, and R. S. Shenoy, "Overview of Candidate Device Technologies for Storage-Class Memory," *IBM Journal of Research and Development*, vol. 52, no. 4.5, pp. 449–464, July 2008.

[12] H. S. P. Wong, H. Y. Lee, S. Yu, Y. S. Chen, Y. Wu, P. S. Chen, B. Lee, F. T. Chen, and M. J. Tsai, "Metal-Oxide RRAM," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, June 2012.

[13] P. E. Gaillardon, M. H. Ben-Jamaa, G. B. Beneventi, F. Clermidy, and L. Perniola, "Emerging Memory Technologies for Reconfigurable Routing in FPGA Architecture," in *2010 17th IEEE International Conference on Electronics, Circuits and Systems*, Dec 2010, pp. 62–65.

[14] P. E. Gaillardon, D. Sacchetto, G. B. Beneventi, M. H. B. Jamaa, L. Perniola, F. Clermidy, I. O'Connor, and G. D. Micheli, "Design and Architectural Assessment of 3-D Resistive Memory Technologies in FPGAs," *IEEE Transactions on Nanotechnology*, vol. 12, no. 1, pp. 40–50, Jan 2013.

[15] X. Tang, P. E. Gaillardon, and G. D. Micheli, "A High-Performance Low-Power Near-V_t RRAM-based FPGA," in *2014 International Conference on Field-Programmable Technology (FPT)*, Dec 2014, pp. 207–214.

[16] ——, "Accurate Power Analysis for Near-Vt RRAM-based FPGA," in *2015 25th International Conference on Field Programmable Logic and Applications (FPL)*, Sept 2015, pp. 1–4.

[17] X. Tang, E. Giacomin, G. D. Micheli, and P. E. Gaillardon, "Circuit Designs of High-Performance and Low-Power RRAM-Based Multiplexers Based on 4T(ransistor)1R(RAM) Programming Structure," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 5, pp. 1173–1186, May 2017.

[18] J. R. V. Betz and A. Marquardt, *Architecture and CAD for Deep-Submicro FPGAs*. Kluwer Academic Publishers Norwell, MA, USA, 1999.

[19] Intel Corporation. (2017) Stratix 10 GX/SX Device Overview. [Online]. Available: https://www.altera.com/documentation/joc1442261161666.html#joc1443027925492

[20] Xilinx Inc. (2017) All Programmable 7 Series Product Selection Guide (XMP101). [Online]. Available: https://www.xilinx.com/support/documentation/selection-guides/7-series-product-selection-guide.pdf

[21] J. Greene, S. Kaptanoglu, W. Feng, V. Hecht, J. Landry, F. Li, A. Krouglyanskiy, M. Morosan, and V. Pevzner, "A 65nm Flash-based FPGA Fabric Optimized for Low Cost and Power," in *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays (FPGA '11)*. New York, NY, USA: ACM, 2011, pp. 87–96.

[22] M. Hutton, J. Schleicher, D. Lewis, B. Pedersen, R. Yuan, S. Kaptanoglu, G. Baeckler, B. Ratchev, K. Padalia, M. Bourgeault *et al.*, "Improving FPGA Performance and Area Using an Adaptive Logic Module," *Field Programmable Logic and Application*, pp. 135–144, 2004.

[23] J. Rose, J. Luu, C. W. Yu, O. Densmore, J. Goeders, A. Somerville, K. B. Kent, P. Jamieson, and J. Anderson, "The VTR Project: Architecture and CAD for FPGAs from Verilog to Routing," in *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '12. New York, NY, USA: ACM, 2012, pp. 77–86. [Online]. Available: http://doi.acm.org/10.1145/2145694.2145708

[24] Berkeley Logic Synthesis and Verification Group. ABC: A System for Sequential Synthesis and Verification.

[25] J. Lamoureux and S. J. E. Wilton, "Activity Estimation for Field-Programmable Gate Arrays," in *2006 International Conference on Field Programmable Logic and Applications*, Aug 2006, pp. 1–8.

[26] X. Tang, P. E. Gaillardon, and G. D. Micheli, "FPGA-SPICE: A Simulation-based Power Estimation Framework for fpgas," in *2015 33rd IEEE International Conference on Computer Design (ICCD)*, Oct 2015, pp. 696–703.

[27] X. Tang, E. Giacomin, G. Micheli, and P.-E. Gaillardon, "FPGA-SPICE: A Simulation-based Architecture Evaluation Framework for FPGAs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, p. Accept to publication, 2018, DOI: 10.1109/TVLSI.2018.2883923.

[28] Synoposys Inc. (2010) HSPICE: The Gold Standard for Accurate Circuit Simulation. [Online]. Available: https://www.synopsys.com/content/dam/synopsys/verification/datasheets/hspice-ds.pdf

[29] H. Y. Lee, P. S. Chen, T. Y. Wu, Y. S. Chen, C. C. Wang, P. J. Tzeng, C. H. Lin, F. Chen, C. H. Lien, and M. . Tsai, in *2008 IEEE International Electron Devices Meeting*.

[30] X. Tang, E. Giacomin, G. D. Micheli, and P. Gaillardon, "Post-p amp;r performance and power analysis for rram-based fpgas," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 3, pp. 639–650, Sept 2018.

[31] Z. Jiang, S. Yu, Y. Wu, J. H. Engel, X. Guan, and H.-S. P. Wong, "Verilog-A Compact Model for Oxide-based Resistive Random Access Memory (RRAM)," in *Simulation of Semiconductor Processes and Devices (SISPAD), 2014 International Conference on*. IEEE, 2014, pp. 41–44.

[32] C. Chiasson and V. Betz, "COFFE: Fully-Automated Transistor Sizing for FPGAs," in *2013 International Conference on Field-Programmable Technology (FPT)*, Dec 2013, pp. 34–41.

[33] G. Lemieux, E. Lee, M. Tom, and A. Yu, "Directional and Single-Driver Wires in FPGA Interconnect," in *Proceedings. 2004 IEEE International Conference on Field- Programmable Technology (IEEE Cat. No.04EX921)*, Dec 2004, pp. 41–48.

[34] S. Yang, *Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0*. Microelectronics Center of North Carolina (MCNC), 1991.